

What is claimed is:

1. An alarm management system, comprising:
 - a hierarchical database of alarm source identifiers, each alarm source identifier being associated with a corresponding software entity capable of generating an alarm;
 - an alarm processor, the alarm processor receiving an alarm from one of the software entities, invoking a corresponding alarm controller for the one of the software entities, accessing the hierarchical database to identify a parent software entity of the one of the software entities, and invoking a corresponding alarm controller for the parent software entity.
2. The alarm management system of claim 1, further comprising an alarm manager, wherein the alarm processor invokes the corresponding alarm controller for the one of the software entities, accesses the hierarchical database to identify the parent software entity of the one of the software entities, and invokes the corresponding alarm controller for the parent software entity via the alarm manager.
3. The alarm management system of claim 1, wherein the alarm includes an alarm source identifier.
4. The alarm management system of claim 1, wherein the alarm processor, upon receiving the alarm, creates a time stamp for the alarm.
5. The alarm management system of claim 3, wherein the alarm source identifier is unique to the software entity generating the alarm during a lifetime of the alarm.
6. The alarm management system of claim 3, wherein the alarm includes an alarm state, the alarm state being one of an alarm cleared and an alarm set.

7. The alarm management system of claim 1, wherein the alarm includes an AlarmId, timeStamp, SourceId, state, type, ProbableCause, and perceivedSeverity in accordance with ITU X.733.

8. The alarm management system of claim 6, wherein the alarm further includes a specificProblem in accordance with ITU X.733.

9. The alarm management system of claim 1, wherein each alarm controller is a software entity capable of generating an alarm, and the hierarchical database includes an alarm source identifier corresponding to the alarm controller.

10. The system of claim 1, wherein the alarm processor invokes the corresponding alarm controller for the parent software entity regardless of whether the alarm was cleared by the one of the software entities.

11. The system of claim 1, wherein the alarm processor does not invoke the parent software entity if the alarm was cleared by the one of the software entities.

12. The system of claim 1, wherein the alarm processor accesses the hierarchical database before it invokes the corresponding alarm controller for the one of the software entities.

13. The system of claim 1, wherein the alarm processor accesses the hierarchical database after it invokes the corresponding alarm controller for the one of the software entities.

14. An alarm management system, comprising:

- a hierarchical database of alarm source identifiers, each alarm source identifier being associated with a corresponding software entity capable of generating an alarm;
- a corresponding alarm hook for each software entity;

a plurality of alarm handlers, each software entity being associated with at least one of the alarm handlers;

an alarm processor, the alarm processor receiving an alarm from one of the software entities and invoking the corresponding alarm hook,
invoking the at least one alarm handler for the one of the software entities,; and
accessing the hierarchical database to identify a parent software entity of the one of the software entities based upon the alarm source identifier associated with the one of the software entities, and
invoking the at least one alarm handler associated with the parent software entity.

15. The alarm management system of claim 14, wherein the alarm includes an alarm source identifier.

16. The alarm management system of claim 15, wherein the alarm processor, upon receiving the alarm, creates a time stamp for the alarm.

17. The alarm management system of claim 15, wherein the alarm includes an alarm identifier, the alarm identifier being unique to the software entity generating the alarm during a lifetime of the alarm.

18. The alarm management system of claim 15, wherein the alarm includes an alarm state, the alarm state being one of an alarm cleared and an alarm set.

19. The alarm management system of claim 14, wherein the alarm includes an AlarmId, timeStamp, SourceId, state, type, ProbableCause, and perceivedSeverity in accordance with ITU X.733.

20. The alarm management system of claim 19, wherein the alarm further includes a specificProblem in accordance with ITU X.733.

21. A method for responding to an alarm comprising:

- receiving an alarm from one of a plurality of software entities,
- identifying a parent software entity of the one of the software entities,
- invoking a corresponding alarm controller for the one of the software entities;
- and invoking a corresponding alarm controller for the parent software entity.

22. The method of claim 21, wherein the step of identifying the parent software entity further comprises

- accessing a hierarchical database of alarm source identifiers, each alarm source identifier being associated with a corresponding software entity of the plurality of software entities; and
- identifying the parent software entity based upon the alarm source identifier associated with the one of the software entities.

23. The method of claim 22, wherein the alarm includes an alarm source identifier.

24. The method of claim 22, further comprising the step of, upon receiving the alarm, creating a time stamp for the alarm.

25. The method of claim 23, wherein the alarm includes an alarm identifier, the alarm identifier being unique to the software entity generating the alarm during a lifetime of the alarm.

26. The method of claim 23, wherein the alarm includes an alarm state, the alarm state being one of an alarm cleared and an alarm set.

27. The method of claim 22, wherein the alarm includes an AlarmId, timeStamp, SourceId, state, type, ProbableCause, and perceivedSeverity in accordance with ITU X.733.

28. The method of claim 27, wherein the alarm further includes a specificProblem in accordance with ITU X.733.

29. The method of claim 21, wherein the step of invoking a corresponding alarm controller for the one of the software entities is performed before the identifying step.

30. The method of claim 21, wherein the step of invoking a corresponding alarm controller for the present software entity is not performed if the alarm is cleared by the one of the plurality of software entities.

31. A method for responding to an alarm comprising:

- receiving an alarm from one of a plurality of software entities and invoking a corresponding alarm hook;
- invoking at least one alarm handler associated with the one of the software entities;
- accessing a hierarchical database of alarm source identifiers, each alarm source identifier being associated with a corresponding software entity of the plurality of software entities;
- identifying a parent software entity of the one of the software entities based upon the alarm source identifier associated with the one of the software entities; and
- invoking at least one alarm handler associated with the parent software entity.

32. The method of claim 31, wherein the step of receiving an alarm further comprises associating the alarm with the one of the plurality of software entities by accessing the hierarchical database.

33. The method of claim 31, wherein the alarm includes an alarm source identifier.

34. The method of claim 31, further comprising the step of, upon receiving the alarm, creating a time stamp for the alarm.

35. The method of claim 33, wherein the alarm includes an alarm identifier, the alarm identifier being unique to the software entity generating the alarm during a lifetime of the alarm.

36. The method of claim 33, wherein the alarm includes an alarm state, the alarm state being one of an alarm cleared and an alarm set.

37. The method of claim 31, wherein the alarm includes an AlarmId, timeStamp, SourceId, state, type, ProbableCause, and perceivedSeverity in accordance with ITU X.733.

38. The method of claim 37, wherein the alarm further includes a specificProblem in accordance with ITU X.733.

39. The method of claim 31, wherein the step of invoking further comprising applying filtering criteria to the alarm and invoking the at least one alarm handler associated with the one of the software entities, and the at least one alarm handler associated with the parent software entity, if the filtering criteria are satisfied.

40. The method of claim 39, wherein the filtering criteria applied to the at least one alarm handler associated with the one of the software entities is different than the filtering criteria applied to the at least one alarm handler associated with the parent software entity.

41. An alarm management system, comprising:

a plurality of software entities capable of generating an alarm;

a hierarchical database of alarm source identifiers, each alarm source identifier being associated with a corresponding one of the software entities;

a plurality of alarm hooks, each alarm hook being associated with one of the software entities;

a plurality of alarm handlers, each alarm handler being associated with one or more of the software entities; and

an alarm processor, the alarm processor

receiving an alarm from one of the software entities and invoking any alarm hook associated with the one of the software entities,

invoking any alarm handlers associated with

accessing the hierarchical database to identify a parent software entity of the one of the software entities based upon the alarm source identifier associated with the one of the software entities; and

invoking any alarm handlers associated with the parent software entity.

42. The alarm management system of claim 41, wherein the alarm includes an alarm source identifier.

43. The alarm management system of claim 42, wherein the alarm processor, upon receiving the alarm, creates a time stamp for the alarm.

44. The alarm management system of claim 42, wherein the alarm includes an alarm identifier, the alarm identifier being unique to the software entity generating the alarm during a lifetime of the alarm.

45. The alarm management system of claim 42, wherein the alarm includes an alarm state, the alarm state being one of an alarm cleared and an alarm set.

46. The alarm management system of claim 41, wherein the alarm includes an AlarmId, timeStamp, SourceId, state, type, ProbableCause, and perceivedSeverity in accordance with ITU X.733.

47. The alarm management system of claim 46, wherein the alarm further includes a specificProblem in accordance with ITU X.733.

48. The system of claim 41, wherein the alarm processor
accesses the hierarchical database to identify a parent software entity of the parent software entity based upon the alarm source identifier associated with the parent software entity, and
invokes any alarm handlers associated with the parent software entity of the parent software entity of the one of the software entities.

49. The method of claim 31, further comprising the steps of
identifying a parent software entity of the parent software entity of the one of the software entities based upon the alarm source identifier associated with the parent software entity of the parent software entity of the one of the software entities; and
invoking the at least one alarm handler associated with the parent software entity of the parent software entity of the one of the software entities.

50. The system of claim 41, wherein the alarm processor
accesses the hierarchical database to identify each ancestor software entity of the parent software entity, and
invokes any alarm handlers associated with each ancestor software entity of the parent software entity of the one of the software entities.

51. The method of claim 31, further comprising the steps of
identifying each ancestor software entity of the parent software entity of the one of the software entities; and

invoking the at least one alarm handler associated with each ancestor software entity of the parent software entity of the one of the software entities.

52. The system of claim 1, wherein the alarm processor accesses the hierarchical database to identify each ancestor software entity of the parent software entity of the one of the software entities, and invokes any alarm controllers corresponding each ancestor software entity.
53. The system of claim 1, further comprising a processor, wherein the alarm processor executes on the processor.
54. A computer readable medium, having stored thereon, computer executable process steps operative to control a computer, the computer executable process steps comprising:
 - receiving an alarm from one of a plurality of software entities,
 - identifying a parent software entity of the one of the software entities,
 - invoking a corresponding alarm controller for the one of the software entities;
 - and invoking a corresponding alarm controller for the parent software entity.
55. The computer readable medium of claim 54, wherein the step of identifying the parent software entity further comprises
 - accessing a hierarchical database of alarm source identifiers, each alarm source identifier being associated with a corresponding software entity of the plurality of software entities; and
 - identifying the parent software entity based upon the alarm source identifier associated with the one of the software entities.
56. The computer readable medium of claim 55, wherein the alarm includes an alarm source identifier.

57. The computer readable medium of claim 55, further comprising the step of, upon receiving the alarm, creating a time stamp for the alarm.

58. The computer readable medium of claim 56, wherein the alarm includes an alarm identifier, the alarm identifier being unique to the software entity generating the alarm during a lifetime of the alarm.

59. The computer readable medium of claim 56, wherein the alarm includes an alarm state, the alarm state being one of an alarm cleared and an alarm set.

60. The computer readable medium of claim 55, wherein the alarm includes an AlarmId, timeStamp, SourceId, state, type, ProbableCause, and perceivedSeverity in accordance with ITU X.733.

61. The computer readable medium of claim 60, wherein the alarm further includes a specificProblem in accordance with ITU X.733.

62. The computer readable medium of claim 54, wherein the step of invoking a corresponding alarm controller for the one of the software entities is performed before the identifying step.

63. The computer readable medium of claim 54, wherein the step of invoking a corresponding alarm controller for the present software entity is not performed if the alarm is cleared by the one of the plurality of software entities.

64. A computer readable medium, having stored thereon, computer executable process steps operative to control a computer, the computer executable process steps comprising:

receiving an alarm from one of a plurality of software entities and invoking a corresponding alarm hook;

invoking at least one alarm handler associated with the one of the software entities;

accessing a hierarchical database of alarm source identifiers, each alarm source identifier being associated with a corresponding software entity of the plurality of software entities;

identifying a parent software entity of the one of the software entities based upon the alarm source identifier associated with the one of the software entities; and
invoking at least one alarm handler associated with the parent software entity.

65. The computer readable medium of claim 64, wherein the step of receiving an alarm further comprises associating the alarm with the one of the plurality of software entities by accessing the hierarchical database.

66. The computer readable medium of claim 64, wherein the alarm includes an alarm source identifier.

67. The computer readable medium of claim 64, further comprising the step of, upon receiving the alarm, creating a time stamp for the alarm.

68. The computer readable medium of claim 66, wherein the alarm includes an alarm identifier, the alarm identifier being unique to the software entity generating the alarm during a lifetime of the alarm.

69. The computer readable medium of claim 66, wherein the alarm includes an alarm state, the alarm state being one of an alarm cleared and an alarm set.

70. The computer readable medium of claim 64, wherein the alarm includes an AlarmId, timeStamp, SourceId, state, type, ProbableCause, and perceivedSeverity in accordance with ITU X.733.

71. The computer readable medium of claim 70, wherein the alarm further includes a specificProblem in accordance with ITU X.733.

72. The computer readable medium of claim 64, wherein the step of invoking further comprising applying filtering criteria to the alarm and invoking the at least one alarm handler associated with the one of the software entities, and the at least one alarm handler associated with the parent software entity, if the filtering criteria are satisfied.

73. The computer readable medium of claim 72, wherein the filtering criteria applied to the at least one alarm handler associated with the one of the software entities is different than the filtering criteria applied to the at least one alarm handler associated with the parent software entity.